Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Procedia Technology 9 (2013) 282 – 290

Procedia Technology

CENTERIS 2013 - Conference on ENTERprise Information Systems / ProjMAN 2013 - International Conference on Project MANagement / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies

Web Service Integration within Next Generation CEMIS

Tariq Mahmoud*, Barbara Rapp, Sebastian van Vliet

Carl von Ossietzky University of Oldenburg, Ammerländer Heerstr. 114-118, 26129 Oldenburg, Germany

Abstract

This paper presents an implementation of the Corporate Environmental Management Information Systems (CEMIS) Next Generation platform (the core component) from the IT-for-Green research project. Key personnel from the research and transfer network ERTEMIS (European Research and Transfer network for Environmental Management Information Systems) work in the IT-for-Green project in order to allow companies and their processes becoming more environmental-friendly by means of information processing. In the frame of this project, the CEMIS Next Generation will be developed, to integrate research concepts of current interest and investigate their feasibility through a prototypical implementation. In this paper we focus on two major components: workflow management system implemented using State Chart XML (SCXML) and green service mall realized as set of standardized Web Services.

Selection and/or peer-review under responsibility of SCIKA – Association for Promotion and Dissemination of Scientific Knowledge

Keywords: Web Services, Green Service Mall, Workflows, SCXML, CEMIS.

1. Introduction

The demand for sustainable development requires that decision-makers in all organizational forms are able to take a multitude of heterogeneous data into account. Information systems in this domain are typically

* Corresponding author. Tel.: +49 (0) 441 / 798 - 44 94; fax: +49 (0) 441 / 798 - 44 72.
E-mail address: mahmoud@wi-ol.de.

Selection and/or peer-review under responsibility of SCIKA – Association for Promotion and Dissemination of Scientific Knowledge
doi:10.1016/j.protcy.2013.12.032

Corporate Environmental Management Information Systems (CEMIS) [1]. Currently, used CEMIS are not capable to cope with requirements resulting from the sustainability discussion, because they do not apply strategic decision support at earlier development stages [2] and are commonly used as end-of-pipe-solutions. Factors that influence the development of sustainable products, business processes or holistic models for cause-and-effect chains are still missing. A resource-friendly design of business processes and their energy- and material-efficient are controlling additional demands for sustainability-oriented organizational structures.

To face today's problems, the European research and transfer network for environmental management information systems (ERTEMIS) has been set up. This community initiated the "IT-for-Green: environment, energy and resource management with next generation CEMIS" project in 2011, with the goal to develop a CEMIS that covers the whole product lifecycle. The important components of this CEMIS are reflected in three Modules, namely "Green IT", "Green Logistics" and "Sustainability Reporting and -dialogue" [3].

Additionally a platform that integrates these modules together is designed to be open and extensible for new nodules and services through a workflow-based and service-oriented platform. The rest of this paper is divided as follows: Section 2 presents this platform, Section3 presents the SCXML Workflows within this platform, Section 4 presents its so called Green Service Mall and finally this paper sums up with a short conclusion of what had been presented.

2. Next Generation CEMIS Architecture (Platform Scope)

Next Generation CEMIS is built in a modular way and it follows the Service-Oriented Architecture (SOA) concept. The OASIS SOA Reference Model group defines SOA as follows: "SOA is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations" [4]. Service orientation in the context of Next Generation CEMIS means that the smallest entities are Web Services, combined through workflows in logical and thematic coherent units to accomplish task-oriented workings.

Web technologies are considered as the fundamental elements to achieve integration in Business-to-Business (B2B) environments. One of the first Web enablement means were the application servers [5]. Web Services extend the Web enablement evolution and considered as powerful technologies to achieve integration by promoting the loose coupling of software functionalities and providing well-defined programmatic interfaces [6]. Based on that, in this project the Web Services are predestined through their modular character and orchestration capabilities. In particular, workflow-based orchestration is needed to provide as much flexibility as possible in forming workflows that map well with inner business processes.

2.1. Platform

The core system (platform), that will finally make up the next generation in the CEMIS, like mentioned before, allows loose coupling and bundling of necessary functions; like linking pre-implemented functions in a programming language. A green service mall provides a semantically enriched procurement of CEMIS-functionality for individual embedding into workflows or rather application definitions. Embedding environmental considerations into arbitrary business processes allows an intermixed usage of specific functions from self-hosted services, external service providers and non-environmental services like transforming data into reports such as sustainability reports. Figure 1 below shows our CEMIS Next Generation Architecture.

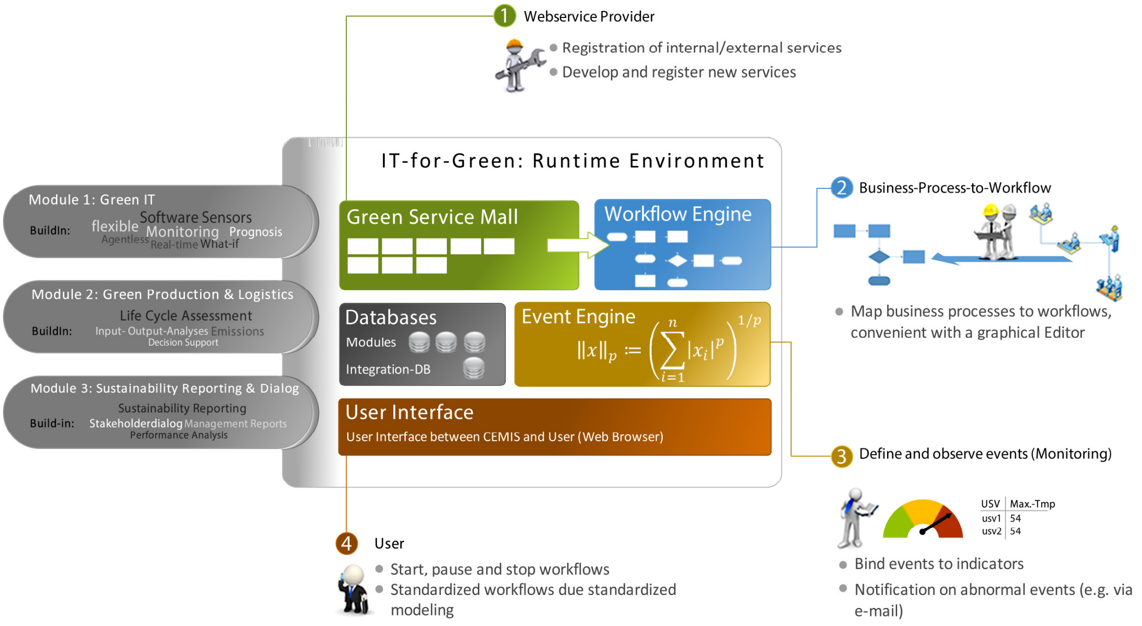

Fig. 1. Next Generation CEMIS Architecture

The runtime environment is the central element of the CEMIS next generation and it is made available on a Web server. As shown in the figure above, the architecture of the CEMIS with a focus on this runtime environment. Primarily, it provides possibilities to integrate new services through Web Service provision (1) besides modeling (2) and execution (4) of workflows. The execution of the workflows is managed by a workflow engine. Using such engine (graphical user interface) enables the internal business processes of the organization to discover and invoke the Green Web Services that are published in the Green Service mall. The workflow editor makes use of the Web Services registered in the Green Service mall regardless of their origin whether they have been created internally in the company or externally. Web service providers can be either external entities or internal providers who develop customized services and make it available by publishing it in the green service mall. The specific metric events within the runtime environment can be defined using the Event Engine (number 3 in the figure above). Temperature excess of a threshold value in a date center is a typical example of such events.

The Green Service mall is the central part of the service-oriented platform. This component provides a set of “yellow pages” and makes them available on the Web. All the services developed in the CEMIS three modules are advertised as services in this mall. The workflow editor can then orchestrate different workflows by discovering, selecting, and invoking these services. In summary, green service mall support all the service phases from discovery till invocation.

The user interface is provided and operated using any kind of Web browser. In particular, the execution of workflows is done via the user interface. Workflows can directly be started, paused and stopped from a Web browser. Another advanced part of the user interface is a dashboard that monitors the defined events besides the Green Service mall (presented as a catalog of services and groups of services classified based on specific criteria).

Authorizations for specific functions and areas can be granted using the rights and roles system that is responsible of issuing rights and roles for services. The rights and role system does not only restrict the access to individual services, rather it hides the information from the unauthorized users (or groups).

3. SCXML Workflows within CEMIS Context

Integration will be achieved by coupling services that encapsulate the actual CEMIS functions by having them called from within an application description that includes not just a simple orchestration of Web Services but rather a user interface together with interaction and issued events that are tied to different interaction elements. Each event may issue changes in control flow, execution of activities, changes (or updates) in the user interface, updates within the shared variables space (the context), etc.

In this way, we are going for a system, that allows for literally programming own CEMIS without limiting ourselves to a specific programming language on the one hand and that eliminates the need for in-depth programming skills when defining a CEMIS on the other. At the same time, a definition as a graph allows for the usage of visual modeling tools. At the same time, the use of a state chart representation with SCXML as definition language allows for interpretation at runtime without a need for a compiler environment [7]. Each service method will have access via the context to the formatting scheme as well as to various converting utilities for proper data handling while still being able to post arbitrary, own (complex) data types. Because the data format will include a semantic as well as a content part (similar to the division of meta-data and content in HTML), annotations for proper interpretation might be included in the exchanged data. The SCXML description format for our workflows will prescribe to use the same format internally (DSL, scripting, etc.) for seamless integration. For import from and export to external software systems, a mediator for translating from one format into the other and vice versa may easily be integrated along the same scheme that integrates other functionality: by using Web Services.

3.1. General Concept

The term SCXML refers to State Chart XML and is an XML based mark-up that allows control abstraction of an application by a state machine notation. Actually, SCXML is a general purpose event-based state machine language [7] to be used for example for high-level dialog control or as a multimodal control language. It is especially furthered for speech processing applications (VoiceXML, voice application meta-language, call-center management language, etc.) but also recommended as a general process control language for other purposes.

In the first line, SCXML is a standard for state chart descriptions. For execution purposes, an environment capable of interpreting SCXML is necessary. We have chosen Commons SCXML as execution environment (http://commons.apache.org/scxml/) because of its lean implementation and the huge potential for own extensions and simple activity implementations. It also follows the recommendation of the workflow reference model as proposed by the workflow management coalition [8]. Commons SCXML abstracts out all execution environment interfaces allowing for - among others - the registration of custom actions, control flow actions, scripting languages or own extensions of the SCXML semantics, if needed .

In this way, workflows defined in SCXML may connect the results from different service invocations in a programmatic way by harnessing integrated scripting and/or expression languages (e.g. Javascript or EL) using a variables space within a context that is shared among (or at least accessible by) all registered services as well as by the engine that may contribute to the function of a so build application. From an interoperability point of view, clearly, data exchange, typification and method parameterization are important to consider. For this purpose, we are going to integrate a common data exchange format based on XML representations into the IT-for-Green approach.

3.2. Activities

One major construct of a workflow system is the concept of an activity that encapsulates principally the actual functionality of the application described by the workflow.

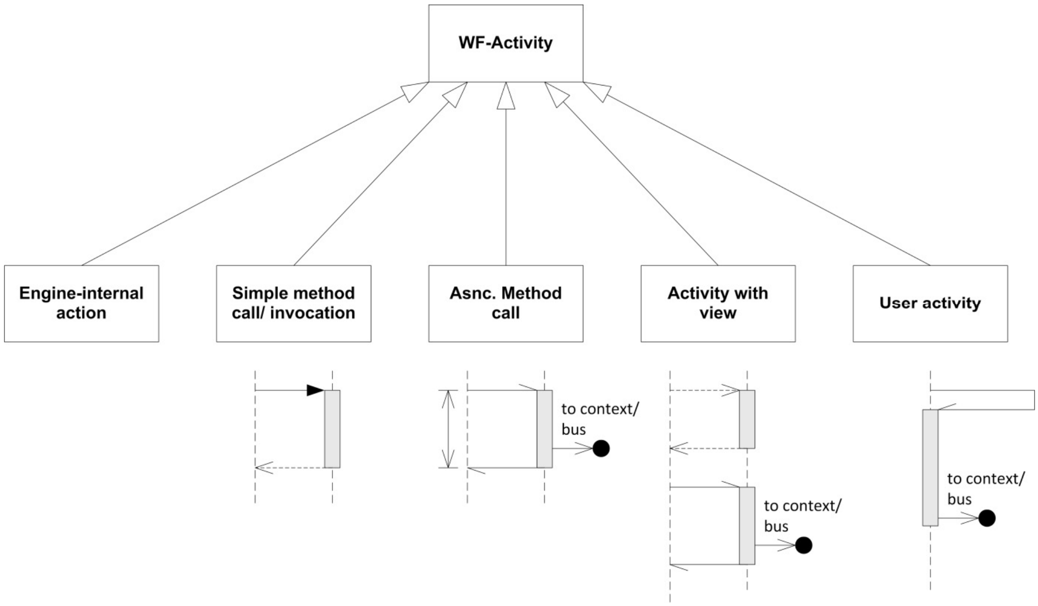

Fig. 2. Main Categories of Activities within the IT-for-Green Workflow Management System

Usually, the basic functionality of a workflow is encapsulated within the implementation activities that are called from the engine during workflow execution. Figure 2 shows the main hierarchy of activities (including their general execution sequence and behavior) that are incorporated in our system.

Other than mostly usual, our workflow definition allows for the integration of arbitrary programming language constructs or DSL elements that might likewise be integrated by a model based development approach. The main hierarchy, as shown in figure 2, groups the activities as follows:

Engine-internal actions

Internal actions are the most basic sort of activity. These control the internal behavior of the workflow engine and do not have any outbound effect. Examples are: The initialization of variables within the context, internal calculations, firing of internal events, etc. These are usually treated as atomic and executed once at a time.

Simple method call

The method call should be understood as a synonym for several types of procedure calls: simple method calls, remote method invocation, several types of web service invocations (we are going to support at least SOAP [9] and Hessian [10] services) as well as calls to internal (available from plug-in libraries) auxiliary methods (a sort of toolkit for basic everyday tasks). These will be implemented as synchronous calls and will not depend on any user interface elements.

Asynchronous method call

Of course, it cannot be guaranteed that each activity completes fast enough for the workflow engine to wait for a return of the method call. An example might be the gathering of sensor data for a longer time horizon. Activities from this group will therefore start the execution asynchronously. This special activity for services that need more time for execution is provided as a specialized service invocation activity that calls the respective method asynchronously. Asynchronously means that the calling method from the Workflow Management System (WfMS) does not wait for an answer but immediately returns and becomes idle (or free for other activities that may be done in parallel); no active waiting is necessary. On the other hand, this means that the service implementation works independently and has to put the result into the context before it signals a work done event to the WfMS. Writing the result to a communication bus would be an appropriate alternative. Our System will be prepared to incorporate different real time conditions into the execution. For example: if the result is not ready at a given due time, it will be discarded. It is the duty of the activity implementation to send a signal back to the workflow engine when work is finished.

Activity with view integration

Some functionality within the modules is likely to be rather complex, composed procedures. For this reason they will have a need for bringing their own user interface into place for proper interaction with the user and because such complex and versatile tasks may hardly be modeled with an external controlled user interface. Editing a sustainability report may serve as an example. An editor for such purpose will for sure be a too complex user interface to be ad hoc defined with the help of the workflow definition. Moreover, it would induce unnecessary additional and error prone work efforts if not pre-implemented and readily available as a component for use. The activity with view integration is the most complex type of activity and has already been described by [11] for the use case of energy aware product design.

User activity

A user activity incorporates the action of a real person. Such activities might be issued in different ways. The easiest form of a user activity would be the direct request of an input from the user in front of the screen during the course of workflow execution. This type can be embedded directly by defining the user interface e.g. as form for input acquisition within the workflow definition and by rendering it. The form would then send the input plus an identifying done event back to the workflow system. Another type of user interaction would be messaging some different user within a collaboration scenario to do something asynchronously. Such sort of interaction will require the other user to confirm his work.

Language related activities

Some specialized activities will be related to integrate programming structures such as structs for control flow (loops, if-then-else construct), memory access (transient and persistent shared data via context), etc. These activities will make up (together with own scripting extension) a customizable form of built in domain specific language.

3.3. Programming Abilities

The following code fragment shows a very simple example for the interplay of a user activity with a generated form for acquiring input and then post-processing this input by invoking a given service:

```
<state id="input">
        <onentry>
                <itfg:form var="addend" >
                        <input type="text" name="value" label="addend: " />
                        <input type="submit" value="add" />
                </itfg:form>
        </onentry>
        <transition event="formSubmitted" target="add" />
</state>
<state id="add">
        <onentry>
                <itfg:serviceCall serviceID="addservice123" var="sum">
                        <parameter name="a" value="${sum}" />
                        <parameter name="b" value="${addend}" />
                </itfg:serviceCall>
                <itfg:optionPane text="Do you want to continue adding?">
                        <itfg:option text="more" event="moreInput"/>
                        <itfg:option text="done" event="calculationDone"/>
                </itfg:optionPane>
        </onentry>
        <transition event="moreInput" target="input" />
        <transition event="calculationDone" target="createChart" />
</state>
```

In this example two states are shown from a larger example (not fully shown here) that lets a user repeatedly enter values that are added up by a service. Depending on the users decisions the process is repeated or ends with the generation of a result chart. The input state issues as soon as entered the generation of an input dialog that is sent to the GUI. As in this state definition constituted, entering data will issue an event that is sent to the workflow engine and issues a transition to the add state that first invokes a math service (with parameters taken from the context by EL) and then generates a second form that enables the user to alter control flow.

4. Green Service Mall

This section illustrates the internal structure of the IT-for-Green's main Web Service repository component. This component is called the green service mall. It constitutes a part of the typical service-oriented paradigm in which the services from the project's modules are published. In other words, this green service mall serves as the project's yellow pages to provide the system's users and developers with the services' information like endpoints, names, descriptions, etc. As soon as the service supplied by a specific developer in one of the project's modules is published, the service consumers can discover the green service mall to fetch its information and consume it consequently.

Figure 3 shows the internal architecture of the green service mall component. It is hosted by a Web Server that has a unique address. This address is known by all the project modules' developers who are depicted in the Figure below as service providers. These latter are linked to the "registry/look-up" Web green service mall's internal component. The main purpose behind this internal component is to store all service information so that they can be located upon request. Service consumers can be bound to the services supplied by service providers after discovering them in this component.

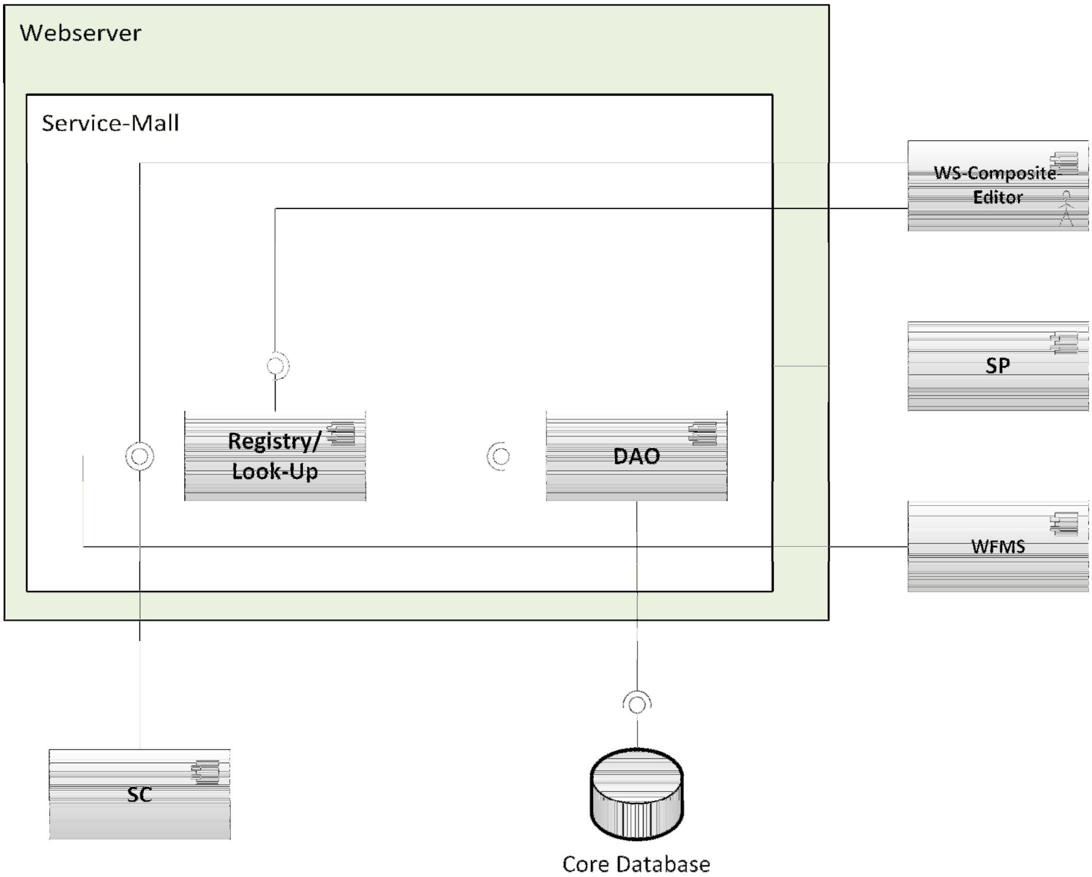

Fig. 3. Component model of the Green Service Mall

The green service mall in this project serves as the main resource of services that are developed by its three main modules. Moreover, external service providers can supply their services and publish them in this mall if they have the suitable rights and privileges. Since most of the activities in this project's workflows are realizes as Web Services, all the information requires to locate and eventually call these services resides in this mall. As a consequence, the green service mall represents the key Web Service integration medium via which the project's WfMS requests the set of required services to execute the project's workflows.

5. Conclusion

The Next Generation CEMIS demands for a way defining environmental management related applications on demand from an easily available and externally (by experts) updated toolbox of remote services bundled together with in-house processes and data. We presented our first consideration regarding a new way of implementing future CEMIS with the help of SCXML. Such definition will allow defining workflows as executable state machine definition with the ability of integrating different DSL and scripting languages as well as extensible mark-up definitions and activity descriptions. A standardized, common data format accompanies the workflow description. In this way, such a system will allow for an easy implementation by definition approach of future highly integrative and tailor made CEMIS systems. Moreover, this paper presented the internal structure of the green service mall that serves as a Web Service repository for the whole project. Eventually, the final prototypical implementations of this project will prove its applicability in the environmental information systems domain.

Acknowledgements

These and the Reference headings are in bold but have no numbers. Text below continues as normal.

References

[1] Teuteberg, F., Marx Gómez, J., 2010. Corporate Environmental Management Information Systems: Advancements and Trends.
[2] Junker, H., 2010. Die Beliebigkeit der Nachhaltigkeit in der betrieblichen Umweltinformatik. HMD - Praxis der Wirtschaftsinformatik: Green Computing & Sustainability. 274|45, pp. 4–5.
[3] Rapp, B., Solsbach, A., Mahmoud, T., Memari, A., Bremer, J., 2011. IT-for-Green: Next Generation CEMIS for Environmental, Energy and Resource Management. EnviroInfo 2011 - Innovations in Sharing Environmental Observations and Information. pp. 573–581. Shaker Verlag, Ispra, Italy.
[4] MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R., 2006. Reference Model for Service Oriented Architecture 1.0, docs.oasis-open.org/soa-rm/v1.0/soa-rm.doc.
[5] Oberle, D., 2005. Semantic Management of Middleware. Springer. 2006 edition (January 4, 2006).
[6] McIlraith, S.A., Son, T.C., Zeng, H., 2001. Semantic Web Services. Intelligent Systems IEEE. 16, 46–53.
[7] Barnett, J., Akolkar, R., Auburn, R., Bodell, M., Burnett, D.C., Carter, J., McGlashan, S., Lager, T., Helbing, M., Hosn, R., others, 2009. State Chart XML (SCXML): State Machine Notation for Control Abstraction. World Wide Web Consort. Working Draft: Wd-Scxml-20090507, May 2009.
[8] Hollingsworth, D., 1995. Workflow Management Coalition: The Workflow Reference Model. Workflow Management Coalition.
[9] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S., Winer, D., 2000. Simple Object Access Protocol (SOAP) 1.1.
[10] Caucho Technology, 2012. Hessian Binary Web Service Protocol, http://hessian.caucho.com/.
[11] Bremer, J., Mahmoud, T., Rapp, B., 2012. Implementing CEMIS Workflows with State Chart XML, in "EnviroInfo 2012 - Part 2: Open Data and Industrial Ecological Management" Arndt, H.K., Knetsch, G., and Pillmann, W., Editors. pp. 749-757. Shaker Verlag, Dessau, Germany.